

Many Faces of a Tree

Simon Urbanek

Simon.Urbanek@math.uni-augsburg.de

Department of Computer Oriented Statistics and Data Analysis,
University of Augsburg, Germany

Interface 2003

Abstract

Tree based models represent an appealing alternative to other commonly used methods for classification and regression tasks. Beside their statistical properties one of the most valued features of tree models is the straight-forward interpretability. Tree models can be often easily communicated to the data set owner who may have less profound statistical knowledge. In order to properly analyze and interpret a tree model, it is important to look at the model from different views and consider the underlying data. In this paper we will present several ways of looking at a tree model, some of them less known. Unfortunately most software tools used for tree model building lack the features necessary for visualization and deeper analysis of the models. In this paper we will also briefly illustrate how our research software, KLIMT (Klassifikation - Interactive Methods for Trees), can be used for interactive graphical analysis of trees using these extended visualization methods.

1 Introduction

Statistical tree models based on recursive partitioning of the covariates space are successfully used in many different scientific domains. Compared to other models they are versatile, because they achieve satisfactory results for non-linear problems, they have the ability to handle data with missing values and allow less restrictive distribution assumptions. One of the most important properties for practical data analysis is the fact that resulting hierarchical structures are easy to interpret. Often it is necessary to communicate the results to the data set owner, in order to assess the practical value of the proposed model. For many models direct interpretation of model parameters is not possible. Although tree models represent quite a complex model structure, they are easy to interpret.

Since their introduction in statistics by Breiman et al. [1984], the modeling methods have been refined and extended. Many different approaches to pruning, constructing and assessing tree models have been proposed. Today it is fairly easy to grow many trees using various algorithms, such as the Bayesian approach by Chipman et al. [1998] or various modifications of the original

CART method such as discussed by Ripley [1996]. Although the ways of creating tree models are increasing constantly, less research has been devoted to the actual visualization and analysis of tree models. In order to assess the quality of a model for a specific task or to choose a possibly best model we need to work with the tree, analyze it and explore it. Static tree plots display only one view of a tree, but dependencies or special cases can better be detected with multiple views.

In order to provide tools for more thorough visual analysis of trees we have developed the software KLIMT for exploratory data analysis of tree models. All views and methods discussed below are implemented in that software. Beside the variety of tree views KLIMT supports many interactive features, such as selection, queries, zooming, variation of displays, multiple views, pruning and linked highlighting for all plot types. The software meets most requirements for interactive software as described by Unwin [1999]. In this paper we concentrate on the various views, the interactive abilities of KLIMT are only briefly mentioned here. We encourage readers to consult the KLIMT project pages for more information about interactive aspects.

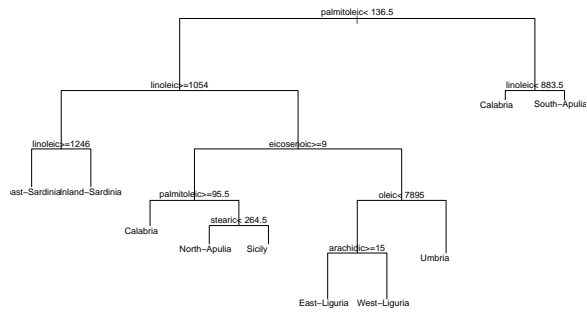
In the first section of this paper we look at the ‘classical’ way of visualizing trees by looking at their hierarchical structure. We illustrate what information can be extracted from different modifications of the view and how the plots can be extended. In the second section we consider different approaches of visualizing tree models. We show strengths and weaknesses of the proposed methods. Finally we conclude with possible future directions of tree visualization, especially concerning multiple models and entire forests.

2 Hierarchical views

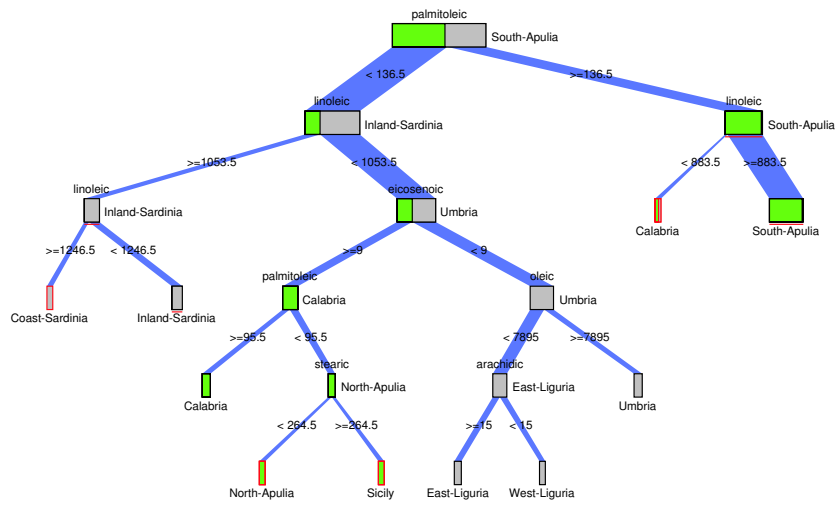
Unlike many statistical plots trees don’t have a fixed or exactly defined graphical representation. Even if we concentrate on the hierarchical structure the variety of different tree plots is huge. Three different ways of drawing the same tree are shown in Fig. 1. The tree was generated using the *rpart* (recursive partitioning) library in R. The underlying dataset consists of 572 cases, where fatty acids of various Italian olive oils were measured. The locations of origin of the oils were grouped into 11 geographic areas of Italy. The goal is to classify the oils into areas based on the acids measurements. Plot A was drawn using R’s native *plot* function, all other plots were created by KLIMT.

Placement of the nodes is one of the most noticeable factors. In some plots the distance between levels of the tree height is constant, in other plots it is proportional to some property of the tree such as the deviance gain in a split (see plot A). It is also possible to display all terminal nodes on the same level to allow better comparison amongst them (plot C). Usually the child nodes are placed below their parents symmetrically, i.e. the centers of parent node and children nodes build an isosceles triangle, but various techniques, such as equidistant partitioning of each level can be used. KLIMT offers various placing algorithms, but also allows the user to freely modify the tree by dragging individual nodes or entire branches.

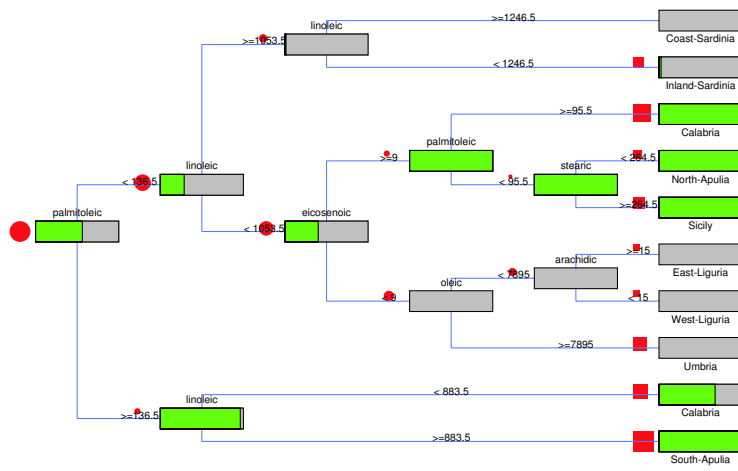
Not only the node placement varies, but also the means of displaying nodes and connecting them. In plot B the width of the connecting lines also represents the amount of cases passed down to the next node, creating an impression of a



Plot A



Plot B



Plot C

Figure 1: Three different ways to visualize the same tree.

'flow' through the tree. Branches with larger population are easily visible at a glance.

It is possible to visualize additional information by using different symbols for nodes, e.g rectangles of various sizes, where the size is proportional to the population of a node (plots B). This allows the use of highlighting to visualize certain groups. In plots B and C oils coming from the southern part of Italy are selected. Plot B allows the global comparison of the subgroups in each node, whereas plot C shows relative proportions in each node. It is clearly visible that except for a few oils coming from Calabria all southern regions can be clearly distinguished from the remaining regions. For more in-depth analysis additional views such as those described in the next section are necessary.

Wilkinson [1999] proposes even more sophisticated horizontal placement of the nodes for his trees called *mobiles*. The placement is chosen with respect to the "statics" of the tree. If the nodes were boxes, the cases marbles, horizontal branches rods and vertical lines wires, the physical model would hang in a plane.

Conventionally trees are plotted in top-bottom orientation, but for large trees this may cause problems because the screen has usually more room in the horizontal than in the vertical direction. KLIMT alternatively allows the tree to be displayed in left-right orientation (see plot C) to avoid this shortcoming.

Overloading plots with information can offset the benefits of the plot, in particular its ability to provide information at a glance. If there is too much information attached to each particular node it is often not possible to display more than two levels of the tree on a screen or a page. Therefore additional tools are necessary to keep track of the overall structure in order not to get lost. Most of these tools, such as zoom, pan, overview window or toggling of labels are available in an interactive context only.

Especially for analysis, visualization of additional information is required. There are basically two possibilities for providing the information: integration of the information in the tree visualization or use of external linked graphics.

Direct integration is limited by the spatial constraints posed by the fixed dimension of a computer screen or other output medium. Its advantage is the immediate impact on the viewer and therefore easier usage. It is recommended to use this kind of visualization for properties that are directly tied to the tree, such as the node size or the criterion used for the tree model construction.

External linked graphics are more flexible, because they are not displayed directly in the tree structure for each node separately, but are only logically linked to a specific node. Spatial constraints are less of a problem because one graphic is displayed instead of many for each node. The disadvantage of linked graphics is that they must be interpreted more carefully. The viewer has to bear in mind the logical link used to construct the graphics as it is not visually attached to its source (node in our case).

KLIMT supports linked highlighting both at case level as well as at node level for linking between trees and plots. Directly related information such as node size or deviance gain are visualized inside the tree structure. In plot C the area of red rectangles represents the remaining deviance in the terminal node. The area of red circles corresponds to the deviance gain achieved by the split in the associated node. This gives a visual guide for the quality of splits and the purity of the nodes.

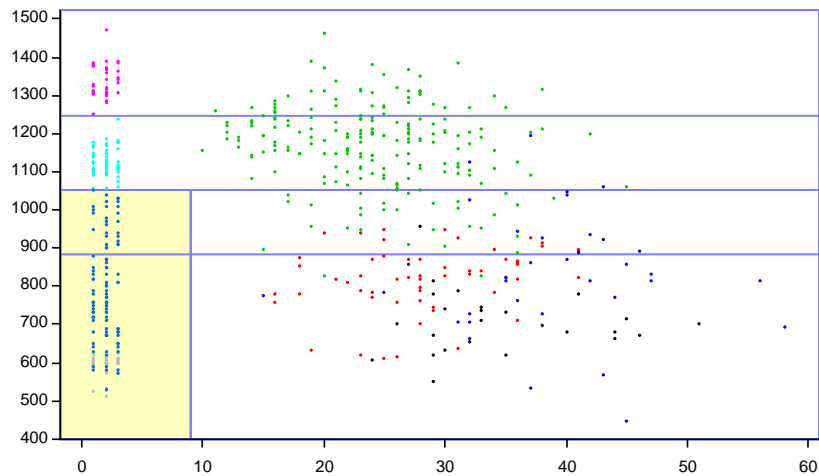


Figure 2: Sectioned scatterplot of linoleic vs eicosenoic acid. Areas of origin are denoted by different colors. The highlighted area corresponds to the currently selected node in the tree.

3 Alternative views

The plot of the hierarchical structure of a tree is not the only way to see the model. A tree model partitions the covariates space according to the splits in the nodes. Univariate splits correspond to partition boundaries orthogonal to the axes. Therefore it is possible to visualize the boundaries in projections of the covariates spaces, such as scatterplots. Examples of such *sectioned scatterplots* are given in Fig. 2 and Fig. 3. Additional lines in the plot correspond to splits in the tree.

The scatterplots were not selected at random, the choice of variables comes directly from the tree. Palmitoleic and linoleic acids appear in the first couple of splits. Eicosenoic acid follows in the sequence and very clearly splits southern regions (all those with $\text{eicosenoic} > 9$, colored green, black, red and dark blue) from others, which can be observed in Fig. 2. The first splits of the tree try to separate the three groups most clearly visible in Fig. 3, namely South Apulia (green), Inland-Sardinia (light blue) and Coast-Sardinia (pink). In addition to the previous plot depth shading was added in Fig. 3. Each partition is filled with a gray shade corresponding to the depth of the corresponding terminal node. Darker shades denote nodes further down the tree. Clearly the area with most mixed distributions is darker since more splits are necessary to distinguish between classes.

Both plots illustrate how the various views explain together the splits of the tree. Although several areas are clearly visible in Fig. 3 the area of Liguria (blue) cannot be distinguished at all, whereas it is easily identified in Fig. 2 (highlighted by yellow background color).

In a special case where only two measurement variables are involved in the decisions, a scatterplot of these two along with partitioning lines describes the entire tree. Each rectangular partition corresponds to a terminal node and

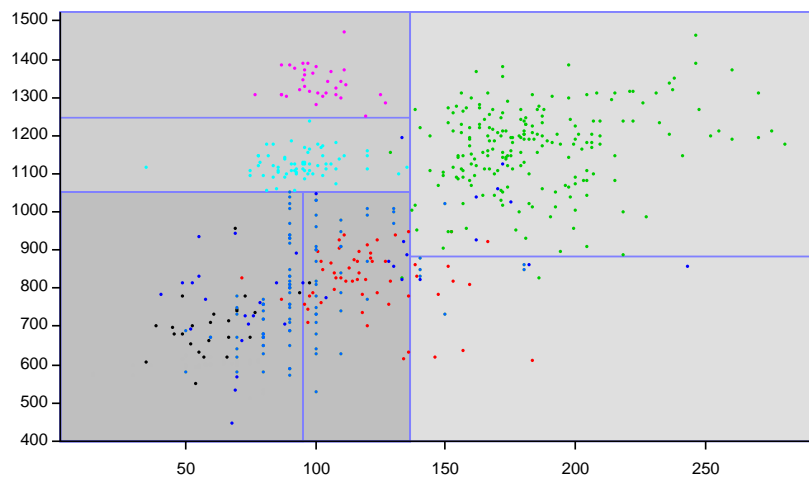


Figure 3: Sectioned scatter plot of linoleic vs palmitoleic acid with additional depth shading.

bears either a class name (for classification trees) or the predicted value (for regression trees). If more than two variables are used in the splitting rules of the tree, each sectioned scatterplot represents only a two-dimensional projection of the measurement space, orthogonal to the axes. Some care must be taken when interpreting such a plot, because splits on variables other than the plotted ones cannot be visualized. Therefore depth-shading, as shown in Fig. 3 can help in such cases. A higher contrast (darker background) to surrounding partitions represents a deeper level in the tree, such as for the two partitions lower left in Fig. 3, and hints at splits which are not visible in the projection.

This approach is especially useful for continuous variables. The scatterplots are helpful when examining individual splits and their neighborhood. Some splits are rather 'sharp' such as the split at $\text{linoleic}=1246.5$, because there are data points very close to the split. Other could be changed quite a lot without affecting the classification, such as the $\text{eicosenoic}=9$ in Fig. 2 which could be placed anywhere between 4 and 14 without changing the result. This can be seen by looking at the neighborhood of the splitting line in the enhanced scatterplot.

Unfortunately the principle cannot be generalized for categorical variables, even in the two-dimensional case, because it is in general not always possible to construct a sequence of categories such that an n -way split of a node results in exactly n continuous partitions of the category space, unless the variable is used only once. This is true for any $n > 1$.

Instead of looking at the measurement space it is possible to consider the number of cases in each node. The corresponding graphic shown in Fig. 4 is structurally similar to a mosaicplot and is often called a *treemap*. The plot is constructed as follows. The basis is a rectangular region representing all cases thus corresponding to the root. For each child of the root the region is partitioned horizontally into pieces proportional to the number of cases in each

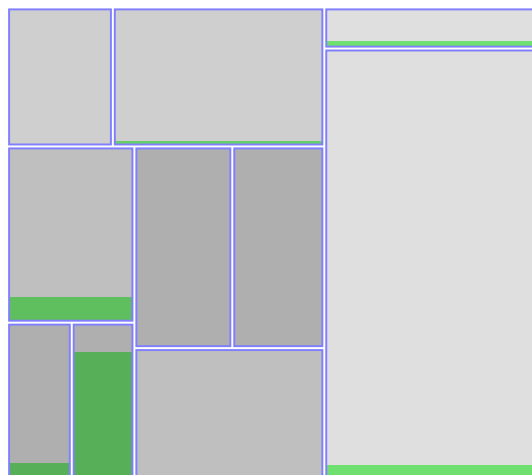


Figure 4: A treemap with depth shading. The are of Sicily is selected.

node. If the node is not terminal, its space is now partitioned vertically according to the size of its children. This procedure is repeated recursively until a leaf is reached while the partitioning direction alternates between horizontal and vertical for each level.

The advantage of treemaps compared to scatterplots is that the limitation of two variables does not apply and even splits on categorical variables can be used. In a two-dimensional, continuous case the corresponding partitions in each plot can be mapped in a bijective fashion, but the area used by the same partition in each plot differs. In a scatter plot the size of a partition is given by the scale of the variables, whereas in a treemap the size is proportional to the number of cases in that partition.

When highlighting is applied, selected cases in the scatterplot are represented by points of a different color and/or size. In a treemap the number of selected cases is proportional to the volume of differently colored area within a partition, usually filled from bottom to top as if water was poured into the partitions. The proportion of the height of such highlighting to the total height of a partition is equal to the proportion of selected cases to the total number of cases in the partition. Therefore treemaps are useful for comparing proportions in the dataset, whereas enhanced scatterplots offer a way to recognize individual points, such as outliers or points at the edge of a split.

The oils coming from the region of Sicily are selected Fig. 4. In fact this region is the hardest to classify, because the oils are very similar to those of other regions in southern Italy. It is easily possible to identify partitions where Sicilian oil is incorrectly classified. Corresponding nodes in the tree can be found by using interactive queries or linked highlighting.

In order to directly compare leaves it is possible to use special plots that are a combination of treemaps and spineplots. The construction of the plot is done like a treemap where partitioning is not performed in alternate directions, but only in the horizontal direction. The resulting plot resembles a spineplot except that individual spines correspond to terminal nodes of the tree and not classes

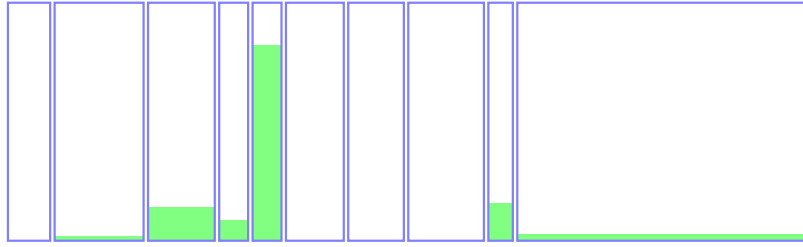


Figure 5: A spineplot of leaves.

of a variable. Therefore we refer to this plot as a *spineplot of leaves* as illustrated in Fig. 5.

This view is especially helpful in conjunction with linked highlighting. The filled area is proportional to the number of cases highlighted in the corresponding terminal node. As in the previous treemap the highlighting represents Sicilian olive oils, allowing visual comparison of the absolute proportions in all nodes. Another property of spineplots is that relative proportions inside each spine correspond to the height of the filled area and hence are directly comparable. This means that both absolute and relative comparisons amongst leaves are possible at a glance.

The disadvantage of both spineplots of leaves and especially treemaps is the fact that identification of a certain node within the plot is somewhat difficult. Direct labeling is not possible, because in general each level of the tree involves different variables in the splits. Interactive queries remain the most appealing solution in this case.

KLIMT implements all three proposed plots. For spineplots of leaves the identification of individual spines is simplified, because the sequences of leaves in the hierarchical tree plot and the corresponding spineplots for leaves are identical.

4 Conclusion

A tree model can be observed from many different angles and each view displays various aspects of the model and the underlying dataset. In order to analyze and interpret a tree model, it is necessary to take a closer look at the model and the data. There are various viewpoints for looking at a tree model: hierarchical structure, splits, associated deviance information, proportions throughout the tree, covariates space or terminal nodes. The hierarchical views are the most popular, but they can still be extended and modified in various ways.

Beside the usual plots emphasizing the hierarchical aspect of a tree, our software KLIMT provides additional views such as sectioned scatterplots, treemaps and spineplots of leaves. All pictures in this paper were generated by KLIMT except for Fig. 1, Plot A. It is important that displays are linked to the data when analyzing a model. The combination of those plots and the interactive features of KLIMT such as hot linking of all views and plots, queries and immediate manipulations of plots, provides an analyst with a versatile tool for

exploratory analysis of individual classification and regression trees. The software is freely available from the project webpage:

<http://www.klimt-project.com/>

Future work will concentrate on visualization of multiple tree models and entire ensembles of trees or forests. Although a single tree is easy to interpret, depending on the data the instability of tree models may question the credibility of a specific tree. Therefore it is necessary to extend the view to multiple potential trees. The goal is to provide interactive tools to assess the justification of interpretability of individual trees even in ensembles of trees.

References

- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- H. Chipman, E. George, and R. McCulloch. Bayesian cart model search (with discussion). *Journal of the American Statistical Association*, 93(443):935–960, Sept. 1998.
- B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
- Antony R. Unwin. Requirements for interactive graphics software for exploratory data analysis. *Computational Statistics*, (14):7–22, 1999.
- L. Wilkinson. *The grammar of graphics*. Springer, 1999.
- KLIMT project site, URL: <http://www.klimt-project.com/>